

BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL

```
BBBBBBBBB      AAAAAA      SSSSSSSS      000000      NN      NN      EEEEEEEEEEE      CCCCCCCC      HH      HH      RRRRRRRR
BBBBBBBBB      AAAAAA      SSSSSSSS      000000      NN      NN      EEEEEEEEEEE      CCCCCCCC      HH      HH      RRRRRRRR
BB      BB      AA      AA      SS      00      00      NN      NN      EE      CC      HH      HH      RR      RR
BB      BB      AA      AA      SS      00      00      NN      NN      EE      CC      HH      HH      RR      RR
BB      BB      AA      AA      SS      00      00      NNNN      NN      EE      CC      HH      HH      RR      RR
BBBBBBBBB      AA      AA      SSSSSS      00      00      NN      NN      EE      CC      HHHHHHHHHH      RRRRRRRR
BBBBBBBBB      AA      AA      SSSSSS      00      00      NN      NN      EE      CC      HHHHHHHHHH      RRRRRRRR
BB      BB      AAAAAAAAAA      SS      00      00      NN      NNNN      EE      CC      HH      HH      RR      RR
BB      BB      AAAAAAAAAA      SS      00      00      NN      NNNN      EE      CC      HH      HH      RR      RR
BB      BB      AA      AA      SS      00      00      NN      NN      EE      CC      HH      HH      RR      RR
BB      BB      AA      AA      SS      00      00      NN      NN      EE      CC      HH      HH      RR      RR
BBBBBBBBB      AA      AA      SSSSSSSS      000000      NN      NN      EEEEEEEEEEE      CCCCCCCC      HH      HH      RR      RR
BBBBBBBBB      AA      AA      SSSSSSSS      000000      NN      NN      EEEEEEEEEEE      CCCCCCCC      HH      HH      RR      RR
```

....
....
....
....

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

```
1 0001 0 MODULE BASONECHR (
2 0002 C IDENT = '1-002'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: BASIC-PLUS-2 Miscellaneous I/O
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the BASIC ONECHR function,
36 0036 1 Which causes the next sequential GET to get only one character.
37 0037 1
38 0038 1 ENVIRONMENT: VAX-11 User Mode
39 0039 1
40 0040 1 AUTHOR: John Sauter, CREATION DATE: 17-APR-1979
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original.
45 0045 1 1-002 - Set up ISB$A_USER_FP. JBS 25-JUL-1979
46 0046 1 --
47 0047 1
48 0048 1 !<BLF/PAGE>
```



```
50 0049 1 |
51 0050 1 | SWITCHES:
52 0051 1 |
53 0052 1 |
54 0053 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
55 0054 1 |
56 0055 1 |
57 0056 1 | LINKAGES:
58 0057 1 |
59 0058 1 |
60 0059 1 | REQUIRE 'RTLIN:OTSLNK';           ! Define linkages
61 0488 1 |
62 0489 1 |
63 0490 1 | TABLE OF CONTENTS:
64 0491 1 |
65 0492 1 |
66 0493 1 | FORWARD ROUTINE
67 0494 1 |     BASSONECHR;                   ! Next GET gets only one character
68 0495 1 |
69 0496 1 |
70 0497 1 | INCLUDE FILES:
71 0498 1 |
72 0499 1 |
73 0500 1 | REQUIRE 'RTLML:OTSLUB';           ! Get LUB definitions
74 0640 1 |
75 0641 1 | REQUIRE 'RTLML:OTISISB';          ! Get ISB definitions
76 0809 1 |
77 0810 1 | REQUIRE 'RTLIN:RTLPSECT';          ! Macros for defining psects
78 0905 1 |
79 0906 1 | LIBRARY 'RTLSTARLE';              ! System symbols
80 0907 1 |
81 0908 1 |
82 0909 1 | MACROS:
83 0910 1 |
84 0911 1 |     NONE
85 0912 1 |
86 0913 1 | EQUATED SYMBOLS:
87 0914 1 |
88 0915 1 |     NONE
89 0916 1 |
90 0917 1 | PSECTS:
91 0918 1 |
92 0919 1 | DECLARE_PSECTS (BAS);              ! Declare psects for BASS facility
93 0920 1 |
94 0921 1 | OWN STORAGE:
95 0922 1 |
96 0923 1 |     NONE
97 0924 1 |
98 0925 1 | EXTERNAL REFERENCES:
99 0926 1 |
100 0927 1 |
101 0928 1 | EXTERNAL ROUTINE
102 0929 1 |     BASS$OPEN_ZERO : NOVALUE,      ! Open channel zero
103 0930 1 |     BASS$CB_PUSH : JSB CB_PUSH NOVALUE, ! Load register CCB
104 0931 1 |     BASS$CB_POP : JSB CB_POP NOVALUE, ! Done with register CCB
105 0932 1 |     BASS$STOP_IO : NOVALUE;         ! Signal fatal I/O error
106 0933 1 |
```

BASSONECHR
1-002

I 16
16-Sep-1984 00:51:55
14-Sep-1984 11:55:24

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BASSONECHR.B32;1

Page 3
(2)

```
: 107      0934 1 !+
: 108      0935 1 !- The following are the error codes used in this module.
: 109      0936 1 !-
: 110      0937 1
: 111      0938 1 EXTERNAL LITERAL
: 112      0939 1      BASSK_IO_CHANOT : UNSIGNED (8);
: 113      0940 1      ! Channel not open.
```

```
115 0941 1 GLOBAL ROUTINE BASSONECHR (
116 0942 1     CHAN
117 0943 1     ) =
118 0944 1
119 0945 1 ++
120 0946 1 FUNCTIONAL DESCRIPTION:
121 0947 1
122 0948 1     Limit the next sequential GET on this channel to a single
123 0949 1     character. This only applies to terminals, and its purpose
124 0950 1     is to permit single-character interaction. It can be used
125 0951 1     in combination with the NOECHO function to allow the BASIC
126 0952 1     program to provide its own line editor.
127 0953 1
128 0954 1 FORMAL PARAMETERS:
129 0955 1
130 0956 1     CHAN.rl.v     The channel to do this to.
131 0957 1
132 0958 1 IMPLICIT INPUTS:
133 0959 1
134 0960 1     NONE
135 0961 1
136 0962 1 IMPLICIT OUTPUTS:
137 0963 1
138 0964 1     LUB$V_ONECHR which, when set, limits the next sequential
139 0965 1     GET to a single character.
140 0966 1
141 0967 1 ROUTINE VALUE:
142 0968 1 COMPLETION CODES:
143 0969 1
144 0970 1     SSS_NORMAL
145 0971 1
146 0972 1 SIDE EFFECTS:
147 0973 1
148 0974 1     Signals if an error is encountered.
149 0975 1     BASS$CB_PUSH will signal if the channel number is invalid.
150 0976 1     We signal BASSK_IO_CHANOT if the channel is not open.
151 0977 1
152 0978 1 --
153 0979 1
154 0980 2 BEGIN
155 0981 2
156 0982 2 BUILTIN
157 0983 2     FP;
158 0984 2
159 0985 2 GLOBAL REGISTER
160 0986 2     CCB = K_CCB_REG : REF BLOCK [, BYTE];
161 0987 2
162 0988 2 LOCAL
163 0989 2     FMP : REF BLOCK [, BYTE];
164 0990 2
165 0991 2     FMP = .FP;
166 0992 2 ++
167 0993 2     Get the CCB for the channel.
168 0994 2 --
169 0995 2
170 0996 2     IF (.CHAN EQL 0)
171 0997 2     THEN
```



```
172 0998 BEGIN
173 0999
174 1000 + The user is referencing his controlling terminal.
175 1001 -
176 1002 BASS$CB_PUSH (LUB$K_LUN_INPU, LUB$K_ILUN_MIN);
177 1003 CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
178 1004
179 1005 + If the controlling terminal is not yet open, open it.
180 1006 -
181 1007
182 1008 IF ( NOT .CCB [LUB$V_OPENED]) THEN BASS$OPEN_ZERO (.FMP [SF$L_SAVE_FP]);
183 1009
184 1010 END
185 1011 ELSE
186 1012 BEGIN
187 1013 +
188 1014 - This is an ordinary channel.
189 1015
190 1016 BASS$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
191 1017 CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
192 1018 END;
193 1019
194 1020 +
195 1021 - If the channel is not now open, either there is a problem with
196 1022 the OPEN code, or the non-zero channel was not first opened.
197 1023
198 1024
199 1025 IF ( NOT .CCB [LUB$V_OPENED]) THEN BASS$STOP_IO (BASS$K_IO_CHANOT);
200 1026
201 1027 +
202 1028 - Now set the ONECHR bit, which will cause the record level code
203 1029 to tell RMS to stop after a single character.
204 1030
205 1031 CCB [LUB$V_ONECHR] = 1;
206 1032 +
207 1033 - We are done with register CCB.
208 1034
209 1035 BASS$CB_POP ();
210 1036 RETURN (TSS$ _NORMAL);
211 1037 END;
```

! end of BASSONECHR

```
.TITLE BASSONECHR
.IDENT \1-002\
```

```
.EXTRN BASS$OPEN_ZERO, BASS$CB_PUSH
.EXTRN BASS$CB_POP, BASS$STOP_IO
.EXTRN BASS$K_IO_CHANOT
```

```
.PSECT _BASS$CODE, NOWRT, SHR, PIC, 2
```

```
.ENTRY BASSONECHR, Save R2,R3,R4,R11
MOVAB BASS$CB_PUSH, R4
MOVL FP, FMP
TSTL CHAN
BNEQ 1$
MNEGL #8, R0
```

```
54 00000000G 00 081C 00000
53 04 5D 9E 00002
AC D0 00009
1E D5 0000C
08 12 0000F
CE 00011
```

```
: 0941
: 0991
: 0996
: 1002
```

52	07	CE	00014	MNEGL	#7, R2	
FF4C	64	16	00017	JSB	BASS\$CB_PUSH	
29	A3	D0	00019	MOVL	12(FMP), -180(CCB)	1003
	AB	E8	0001F	BLBS	-4(CCB), 3\$	1008
00000000G	A3	DD	00023	PUSHL	12(FMP)	
00	01	FB	00026	CALLS	#1, BASS\$OPEN_ZERO	
	0E	11	0002D	BRB	2\$	0996
52	50	D4	0002F	CLRL	R0	1016
	AC	D0	00031	MOVL	CHAN, R2	
FF4C	64	16	00035	JSB	BASS\$CB_PUSH	
0B	A3	D0	00037	MOVL	12(FMP), -180(CCB)	1017
7E	AB	E8	0003D	BLBS	-4(CCB), 3\$	1025
00000000G	8F	9A	00041	MOVZBL	#BASSK IO CHANOT, -(SP)	
A0	01	FB	00045	CALLS	#1, BASS\$STOP_IO	
AB	02	88	0004C	BISB2	#2, -96(CCB)	1031
00000000G	00	16	00050	JSB	BASS\$CB_POP	1035
50	01	D0	00056	MOVL	#1, R0	1036
	04	00	00059	RET		1037

; Routine Size: 90 bytes, Routine Base: _BASS\$CODE + 0000

:	212	1038	1
:	213	1039	1 END
:	214	1040	1
:	215	1041	0 ELUDOM

! end of module BASSONECHR

PSECT SUMMARY

Name	Bytes	Attributes
_BASS\$CODE	90	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	2	0	581	00:01.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD, INITIAL, OPTIMIZE)/NOTRACE/LIS=LIS\$:BASONECHR/OBJ=OBJ\$:BASONECHR MSRC\$:BASONECHR/UPDATE=(ENH\$:BASONECHR

BASSONECHR
1-002

M 16
16-Sep-1984 00:51:55
14-Sep-1984 11:55:24

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BASSONECHR.B32;1

Page 7
(3)

:)

: Size: 90 code + 0 data bytes
: Run Time: 00:08.7
: Elapsed Time: 00:22.4
: Lines/CPU Min: 7212
: Lexemes/CPU-Min: 43461
: Memory Used: 118 pages
: Compilation Complete

0028 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASMD
LIS

BASMDL
LIS

BASNOTIMP
LIS

BASMOVEAR
LIS

BASMSGDEF
LIS

BASMSGGEN
LIS

BASONECHR
LIS

BASMOVE
LIS

BASNUM
LIS

BASNAMEAS
LIS

BASNUM
LIS